UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/696,467	10/28/2003	Yu Kwong Ng	CSCO-7059	6905

7590 01/17/2007

WAGNER, MURABITO & HAO LLP
Third Floor
Two North Market Street
San Jose, CA 95113

EXAMINER	
WALTER, CRAIG E	
ART UNIT	PAPER NUMBER
2188	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	**Application No.** 10/696,467	**Applicant(s)** NG ET AL.
	Examiner Craig E. Walter	**Art Unit** 2188

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 November 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20 and 26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20 and 26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Status of Claims

1. Claims 1-20 and 26 are pending in the Application.

Claims 21-25 remain cancelled.

Claims 11, 7-20 and 26 are amended.

Claims 1-20 and 26 are rejected.

Response to Amendment

2. Applicant's arguments filed on 20 November 2006 in response to the Office Action mailed on 17 August 2006 have been fully considered but they are not persuasive. Therefore, the rejections made in the previous Office Action are maintained, and restated below, with changes as needed to address the amendments.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 7-14 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 7 recites the limitation "the original key" in the final line of the claim. There is insufficient antecedent basis for this limitation in the claim. More specifically, it is unclear which of the plurality of original keys as previously set forth within the claim is being claimed here.

Claims 8-14 are rejected for inheriting the deficiency of claim 7.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 7-10 are rejected under 35 U.S.C. 102(e) as being anticipated by Brandin et al., hereinafter Brandin (US Patent 6,493,813 B1).

As for claim 7, Brandin teaches an apparatus, comprising:

a memory which stores a plurality of partial keys used to determine hashing conflicts, wherein said plurality of partial keys correspond to a plurality of original keys, and wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys (Fig. 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57). Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a plurality of partial keys (i.e. Fig. 13A – each key is split into two partial keys of equal size). Each set of partial keys (i.e. two 64-bit partial keys) form the constituent elements of, and correspond to, its respective original key (128-bits). The partial keys (64-bit) require less memory than the original keys (128-bits).

Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10(exp)8 keys) – The transform generator determines an address and a confirmer for each key (col. 2, lines 47-48. The information determined from each of the keys (or partial keys as shown in Fig. 13a) is used to prevent the occurrence of collisions (i.e. hashing conflicts) – col. 2 lines 21-30);

a hash function block coupled to a memory that applies any polynomial to a full key and generates a hash value which is used to point to one of the plurality of partial keys stored in the memory wherein the partial keys include saved bits comprising a consecutive, sequential string of bits derived from the original key (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a. – the original key (element 312) is split into partial keys, and the hash function is applied. Additionally, referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements

316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39). Additionally note that addressing and pointer information is stored directly in the memory as per col. 2, line 66 through col. 3, line 11;

As for claim 8, Brandin teaches the apparatus of Claim 7, wherein the memory comprises a 2(exp)N hash table size (referring to Fig. 3, the store table example used (element 50) contains 16 entries (i.e. N=4)).

As for claim 9, Brandin teaches the apparatus of Claim 7, wherein the one of the plurality of partial keys stored in the memory comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value (referring again to Fig. 13a, partial key A (element 316) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key. Since the partial key is half the original key's size, the partial key is equal to the size of the original key minus the hash value – col. 7, lines 14-39).

As for claim 10, Brandin teaches the apparatus of Claim 7, wherein the hash function block comprises a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 11-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Rajski et al., hereinafter Rajski (US PG Publication 2002/0016806 A1).

As for claims 11 and 12, Brandin fails to teach his LFSR as corresponding to either a Fibonacci, or a Galois version.

Rajski however teaches a method for synthesizing linear finite state machines, which includes both the Fibonacci, or a Galois versions – paragraph 0002, lines 17-20 and paragraph 0003, lines 1-4 – both types are described in his teachings.

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Rajski's method for synthesizing linear finite state machines for his own LFSRs. By doing so, Brandin would be able to more efficiently implement his LFSR with fewer levels of logic, and a lower internal fan-out of the circuitry, as taught by Rajski (paragraph 0012, lines 1-18).

6. Claims 1-4 and 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin in further view of Biran (US Patent 6,345,347 B1).

As for claim 1, Brandin teaches a method for hashing, comprising:

storing a plurality of partial keys in memory (Fig 13a illustrates two partial keys corresponding to a plurality of original keys in memory, wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys, and wherein said plurality of partial keys are used to determine hashing conflicts (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57. Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys)). Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a plurality of partial keys (i.e. Fig. 13A – each key is split into two partial keys of equal size). Each set of partial keys (i.e. two 64-bit partial keys) form the constituent elements of, and correspond to, its respective original key (128-bits). The partial keys (64-bit) require less memory than the original keys (128-bits);

applying a hash function to an original key of said plurality of original keys to generate a hash value, wherein said hash function comprises any polynomial (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

accessing the memory according to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading a partial key of said plurality of partial keys from the memory corresponding to the hash value, wherein said hash value is based on said original key (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits;

Note giving the limitation "reading a partial key from the memory that corresponds to said hash value, wherein said hash value is based on said original key" its "broadest reasonable interpretation consistent with the specification" (In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)), reading a partial key may include reading the entire key (which is split into more than one partial key), since the each partial key is inherently read during the step of reading the key in its entirety. The hash value will always be "based on the original key" even if said value is only made up by a portion of the original key (i.e. the portion corresponding to the partial key).

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He fails to teach however, actually comparing the keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claim 15, Brandin teaches an apparatus comprising:

means for storing a plurality of partial keys corresponding to a plurality of original keys in memory, wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys, and wherein said plurality of partial keys are used to determine hashing conflicts (Fig 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57). Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys). Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a plurality of partial keys (i.e. Fig. 13A – each key is split into two partial keys of equal size). Each set of partial keys (i.e. two 64-bit partial keys) form the constituent elements of, and correspond to, its respective original key (128-bits). The partial keys (64-bit) require less memory than the original keys (128-bits);

means for applying a hash function to an original key of said plurality of partial keys to generate a hash value, the hash function comprising any N bit polynomial (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

means for accessing the memory according to the hash value, wherein a position to save comprises any N consecutive bits (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49. Referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements 316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39);

means for reading a partial key from the memory corresponding to the hash value, wherein a size to save comprises (less than or equal to) N bits (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11. The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits);

Note giving the limitation "reading a partial key from the memory that corresponds to said hash value, wherein said hash value is based on said original key" its "broadest reasonable interpretation consistent with the specification" (In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)), reading a partial key may include reading the entire key (which is split into more than one partial key), since the each partial key is inherently read during the step of reading the key in its entirety. The hash value will always be "based on the original key" even if said value is only made up by a portion of the original key (i.e. the portion corresponding to the partial key).

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He also teaches the hash table size as 2(exp)N (Fig. 2, 16 entries are disclosed). He fails to teach however, actually comparing the keys (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

As for claims 2 and 16, Brandin teaches the method of Claim 1 (and apparatus of claim 15), wherein the partial key from the memory corresponding to the hash value includes saved bits comprising a consecutive, sequential string of bits, less than or equal to N, which is part of the original key (referring to Fig. 13a, the original key (element 312) is comprised of two partial keys (elements 316 and 320). The bits in each partial key are stored in a sequential line (based on the key length), each containing less bits than the original key – col. 7, lines 14-39).

As for claims 3 and 17, Brandin teaches the method of Claim 2 (and apparatus of claim 16), wherein the partial key from the memory corresponding to the hash value comprises a number of bits equal to or more than the number of bits of the original key minus the number of bits of the hash value (referring again to Fig. 13a, partial key A (element 316) is input into the LFSR to generate a hash value (transform) which is equal in size to the partial key. Since the partial key is half the original key's size, the partial key is equal to the size of the original key minus the hash value – col. 7, lines 14-39).

As for claims 4 and 18, Brandin teaches the method of claim 1 (and apparatus of claim 15), wherein the hash function is implemented by a linear feedback shift register (Fig. 12, element 312 illustrates the LFSR – col. 7, lines 9-11).

7. Claims 6 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin and Biran as applied to claims 1 and 15 above, and in further view of Ji (US PG Publication 2005/0086363 A1).

As for claims 6 and 20, Brandin teaches the method of Claim 1 (and apparatus of claim 15), further comprising:

reading a result from the memory corresponding to the hash value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49, and the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11).

Brandin fails however to teach forwarding a packet of data according to the result read from the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

8. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claim 13, Brandin fails to teach the apparatus of claim 7 further including a reverse function generator coupled to the memory wherein the reverse function generator generates the original key based on the one of the plurality of partial keys stored in the memory and hash value.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21. Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

9. Claims 5 and 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over the combined teachings of Brandin and Biran as applied to claims 1 and 15 above, and in further view of Bryg et al., hereinafter Bryg (US Patent 6,430,670 B1).

As for claims 5 and 19, Brandin fails to teach the method of Claim 1 (and apparatus of claim 15), further comprising applying a reverse function on the partial key from the memory corresponding to the hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key) and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21.

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

10. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin as applied to claim 7 above, and in further view of Ji.

As for claim 14, Brandin fails to teach the apparatus of claim 7 further comprising a forwarding engine coupled to the memory, wherein the forwarding engine forwards a data packet according to information read from the memory at an address corresponding to the one of the plurality of partial keys stored in the memory.

Ji however teaches a traffic flow management system through a multipath network, which uses a router to forward packets of data. The packets are forwarded in accordance with the information provided to system based on the hash value of the data being forwarded (paragraph 0026, lines 15-20).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Ji's traffic flow management system in order for Brandin to send information referenced by his memory store, as a series of packets. By doing so, Brandin would be able to more efficiently send data referenced by the memory store data, which would in turn improve the load balancing during data transmission (paragraph 008, lines 1-16).

11. Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Brandin in further view of Biran and Bryg.

As for claim 26, Brandin teaches a method for accessing data, comprising:

storing a plurality of partial keys corresponding to a plurality of original keys in memory, wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys, and wherein said plurality of partial keys are used to determine hashing conflicts (Fig 13a illustrates two partial keys (elements 316 and 320). Referring to Fig. 1, the memory management system (20) has three elements including the transform generator, a controller and the memory table (26). The keys (which are split into partial keys as illustrated in Fig. 13a-b), are provided to the transform generator – col. 2, lines 54-57). Referring to Fig. 9, each key is stored as an entry in the memory table (i.e. 10[exp]8 keys). Note Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a plurality of partial keys (i.e. Fig. 13A –

each key is split into two partial keys of equal size). Each set of partial keys (i.e. two 64-bit partial keys) form the constituent elements of, and correspond to, its respective original key (128-bits). The partial keys (64-bit) require less memory than the original keys (128-bits);

applying a function to an original key of the plurality original keys to generate a value (col. 2, lines 54-65 – the transform generator uses polynomial code to generate address and confirmer information (i.e. hash value) for the key – This procedure is applied to partial keys in Fig 13a.);

accessing the memory according to the value (the address and confirmer information (i.e. hash value) is used to locate the data in the memory – col. 2, lines 47-49);

reading the partial key of the plurality of partial keys from the memory corresponding to the value and based on the original key (the controller is used to look up the key's association in the memory table based on the information provided by the hash function (the address and confirmer) – col. 2, line 67 through col. 3, line 11). The entry containing the partial key is read in order to obtain this information. Again, Fig. 13a illustrates that this can be applied to the partial keys if the original key is greater than 64 bits;

Note giving the limitation "reading a partial key from the memory that corresponds to said hash value, wherein said hash value is based on said original key" its "broadest reasonable interpretation consistent with the specification" (In re Hyatt, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)), reading a partial key may include reading the entire key (which is split into more than one partial key), since the each partial key is inherently read during the step of reading the key in its entirety. The hash value will always be "based on the original key" even if said value is only made up by a portion of the original key (i.e. the portion corresponding to the partial key).

Brandin further teaches executing a conflict check by comparing the confirmer of a partial key derived from the confirmer of an incoming full key with the confirmer of a partial key stored in the memory ((col. 2, line 66 through col. 3, line 11) – the first confirmer (derived from the first partial key of the full key) is compared with a stored first confirmer at the first address). He fails to teach however, actually comparing the keys in order to determine which data is accessed (in contrast he teaches comparing the values of hashing results produced by applying the transform generator to the keys).

Biran however teaches a system for address protection using a hardware-defined application key, which in fact directly compares the keys in order to mitigate hashing conflicts (col. 2, lines 58-67 – Biran teaches eliminating the possibility of conflicts occurring by directly comparing the keys (in contrast to Brandin's system of comparing the hashed values of the keys)).

It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Biran's address protection system using a hardware-defined application key in his own system. By including Biran's method of comparing the keys, rather than comparing the translated keys, Brandin would be able to compare keys that correspond uniquely to the appropriate hardware address, hence eliminating the possibility of hashing conflicts. This system could easily be implemented in hardware (i.e. Brandin's controller which is used to compare the translated keys), while minimizing processing overhead – col. 2, lines 58-67).

Brandin further fails to disclose applying a reverse function on the partial key and hash value to generate the original key.

Bryg however teaches an apparatus and method for a virtual hashed page table in which his original hashing function is reversible. The hash index (containing a portion of the key, therefore it itself is a partial key and tag are used to uniquely identify the original translation of the key. This procedure can be reversed by applying the reverse hash function on the hash result and the hash identifiers – col. 8, lines 4-21). Note the hash generator hardware is coupled to the system's memory (Fig. 8, element 131).

Again, It would have been obvious to one of ordinary skill in the art at the time of the invention for Brandin to further implement Bryg's apparatus and method for a virtual hashed page table. By doing so, Brandin would benefit from Bryg's virtual hash translating by utilizing two unique address spaces (either multiple or single hashed page table method) – col. 1, lines 18-28. Bryg's apparatus would provide Brandin with a single architectural virtual hash page table, which supports both methods of virtual addressing. In turn Brandin would benefit by increasing the number of operating systems capable of managing the information, and more efficiently utilize the structure, which in the end would save the end user time and memory as taught by Bryg in col. 2, lines 28-40.

Response to Arguments

12. Applicant's arguments filed on 20 November 2006 have been fully considered but they are not persuasive.

13. With respect to claim 7 (under the heading "35 U.S.C. 102 Rejections"), Applicant asserts, "Brandin does teach "a memory which stores a plurality of partial keys used to determine hashing conflicts, wherein said plurality of partial keys correspond to a plurality of original keys, and wherein storage of said plurality of partial keys requires less memory than storage of said plurality of original keys;"". Applicant continues by summarizing Brandin's invention, and contrasts specific elements of his invention with the instant claim. Applicant further alleges that Bradin's two 64-bit transforms (formed from each of the two partial keys) would not result in improved memory efficiency because the memory required to store them would equal the size of the original key, and concludes, "Brandin directly contradicts "...storage of said plurality of partial keys requires less memory than storage of said plurality of original keys". This argument is not persuasive as Examiner maintains that Brandin in fact does anticipate this limitation, and further asserts that Applicant's argument with respect to the storage area consumed by Brandin's transforms in not commensurate with the scope of the amended claim limitation. As for the latter, the claim limitation requires the storage of the partial keys to require less memory than the original key, not the transforms. Brandin teaches, *inter alia*, a plurality of partial keys, each of which consumes less memory than the original (i.e. partial keys = 64 bits, and the original = 128 bits; see the rejection of claim 7, *supra,* and Fig. 13a). As for the former argument (general allegation that the newly

added claim limitations are not present), Examiner maintains that Brandin teaches a plurality of original keys (col. 2, lines 21-65), each of which contains a plurality of partial keys (i.e. Fig. 13A – each key is split into two partial keys of equal size). Each set of partial keys (i.e. two 64-bit partial keys) form the constituent elements of, and correspond to, its respective original key (128-bits). Each of the partial keys (64-bit) requires less memory than its respective original key (128-bits). Note, additionally the key can be 196 bits, split into three partial keys, 64-bits each (col. 7, lines 39-48). In this instance a plurality of partial keys (i.e. two of the three) still consumes less memory than the original key comprised of 196-bits.

14. With respect to claims 1, 15 and 26, (under the heading "35 U.S.C. 103 Rejections"), Applicant sets forth a similar argument as discussed in paragraph 0013 above. This argument is not persuasive for the reasons set forth in paragraph 0013 of this correspondence.

15. Applicant's argument that all remaining dependant claims are allowable for inheriting allowable subject matter from their respective base claims is rendered moot as Examiner maintains that Brandin either anticipates, or renders obvious in view of the remaining cited art, each currently pending base claim per the discussion and arguments *supra*.

Conclusion

16. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Craig E. Walter whose telephone number is (571) 272-8154. The examiner can normally be reached on 8:30a - 5:00p M-F.

18. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung S. Sough can be reached on (571) 272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

19. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Craig E Walter
Examiner
Art Unit 2188

CEW

HYUNG SOUGH
SUPERVISORY PATENT EXAMINER
1-12-07